Networked Applications NWEN 243

Lab Exercise 4 –Basic Cryptography 5%

Objectives

- Experience thinking about security – in this case basic encryption.
- Write a small program to encode a block of text.
- Write a small program that will allow you to perform frequency analysis over a block of text (we provide).

Requirements

- This lab an individual lab written in C.
- We will be writing programs that you execute from the shell command line.
- You must demonstrate and hand in part 1 in the week starting 9th Sept.
- You must demonstrate and hand in part 2 in the week starting 16th Sept.
- Because part 2 will take a long time – I suggest you write this over the lecture break, do not do Part 1 and Part 2 sequentially, you will not have time!

The Exercise

Being a Cryptographer: basic encoding (this is worth 5pts or 5%)

Choose a direct substitution cypher, you can map this as you wish – provided that:

1. character mappings are static. That is, if a maps to c, it will always map to c (in this execution).
2. the mapping is seeded from a key word.

The most basic cypher of this form is a Caesar cypher – but you may choose to optionally implement something a little more challenging if you desire – as long as it meets the limitations set out above.

You are to write two small C programs that will (1) encode using your key, and (2) decode using the same key, e.g.:

%>cat file.txt | **encode** password | **decode** password > newfile.txt

I should then be able to execute diff, and see no changes, e.g.:

%> diff –i file.txt newfile.txt

Assumptions

1. You may assume text is in ASCII and that you do not need to preserve case (just make it all upper case – note '-i' option to diff above) encode symbols, punctuation, spaces or whitespace in general (i.e just pass it through without

change), e.g.:

"*don't do it*" becomes "*efh'w ef xw*"

2. The key is composed of uppercase alpha chars, with no repeats, e.g.:

"IAMWESO"

Submission

In addition to the demo, you must submit your code using the submission system - submit all source needed to compile.

Part 2 – Being a Cryptanalyst: Frequency Analysis (this is worth 10pts or 10%)

This is due for demo in the lab in the 2 weeks after the break and the code must be submitted by the end of your scheduled lab time in the week beginning the 16th September.

Write a short program in C to perform basic frequency analysis on a 1K block of text. Use the character frequencies (English only) from the lecture. Then within a loop, allow the user to enter better 'guesses' to update the mappings until the document is fully decoded – your program should output the final mappings as well as the decoded text.

%> cracker filein.txt fileout.txt

Submission

In addition to the demo, you must submit your code using the submission system - submit all source needed to compile.